INT. J. SYSTEMS SCI., 1982, VOL. 13, NO. 9, 1019–1029

A procedure for recognition of connected handwritten numerals

S. K. PARUI†, B. B. CHAUDHURI†
and D. DUTTA MAJUMDER†

Syntactic methods for segmentation of connected handwritten numerals and for recognition of individual handwritten numerals are described. In the first part the size of individual numerals present in the same chain of connected numerals is required to be nearly uniform. But the recognition procedure for individual numerals is invariant under shape and size. After segmentation and before recognition handwritten numerals are divided into two groups. Handwritten numerals consisting mainly of curves form one group while those consisting of vertical and horizontal strokes form the other. Recognition procedures for the two groups are different but similar. In both cases, for recognition of a numeral only the information about its border is made use of. The extracted border is in fact a sort of skeleton of the numeral pattern and is stored in one-dimensional strings of eight direction codes. On the basis of these one-dimensional strings certain subpatterns are recognized through some automata. The numeral pattern is ultimately recognized from these subpatterns. The method was implemented on an EC 1033 machine with a success rate of about 90%.

1. Introduction

Automatic recognition of handprinted characters has found wide application in postal and bank services, intelligence and customs departments etc. The handprinted characters are usually the alphabets of a natural language or numerals. Under certain constraints they are nowadays recognized by a number of commercial machines. These machines are usually an OCR type accepting certain character sets for handprinting such as ANSI X3.45-1974 which is constrained not only in shape but also in size. But a machine which can reliably read the natural printing of most people is nowhere in sight. However, recognition of unconstrained handwritten characters is now a subject of research (Focht and Burger 1976, Himmel 1976, Ali and Pavlidis 1977) and one promising tool for solving the problem is the syntactic method (Fu 1974, 1977).

The present paper makes an attempt to recognize unconstrained and connected handwritten numerals on the basis of their syntactic features. The procedure has three steps. In the first step a segmentation algorithm separates individual numeral patterns that are present in the input chain of connected handwritten numerals. Then each individual numeral is separately preprocessed. The pre-processing algorithm first smooths the pattern and then extracts from it a sort of a skeleton which is not the mid-line through the pattern but is only one side of the border (thick lines in Fig. 1). The border thus extracted is in the form of a one-dimensional string of the eight directional codes that are shown in Fig. 2. In the last step subpatterns are recognized

---

Received 13 November 1981.

† Electronics and Communication Sciences Unit, Indian Statistical Institute, Calcutta 700035, India.

0020-7721/82/1309 1019 $03.00 © 1982 Taylor & Francis Ltd

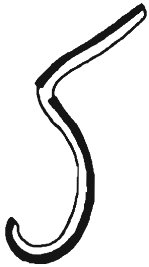

Figure 1. Thick portions on the border represent the skeleton of the pattern.

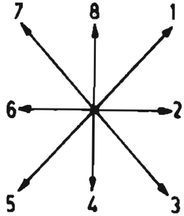

Figure 2. Directional codes.

on the basis of the strings of directional codes and finally the numeral pattern is recognized on the basis of the subpatterns.

Numeral patterns are divided into two overlapping groups after segmentation. The idea of the division stems from the observation that some numerals consist mainly of curves while others consist mainly of horizontal and vertical line segments. In the preprocessing and recognition stages the numerals of the two groups are processed in two different but similar ways which are discussed in §§ 3 and 4 respectively.

2. Segmentation

Input chains of connected handwritten numerals are digitized into two gray levels 0 and 1 which describe the white and black portions of the input respectively. A pixel with gray level 1 will be called a point. For segmentation first the downward protrusions in the input pattern are found (two such protrusions in Fig. 3 (*a*)). Each protrusion means a separate numeral pattern. So the connected portion between two consecutive protrusions has to be disconnected. The algorithm for this starts at the bottom of the left protrusion (point A Fig. 3 (*a*)) and moves along the border in the anticlockwise direction (marked by 2's in Fig. 3 (*a*)). This gives rise to a string of directional codes.

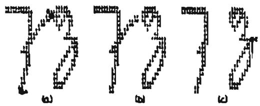

Figure 3. Two connected numerals are being separated.

Point for disconnection is where directional code 1 or 2 is followed by directional code 3 for the first time (point B in Fig. 3 (*a*)). At this point disconnection is done vertically (Fig. 3 (*b*)). The near horizontal portion which is on the left of this point is also removed from the input pattern (Fig. 3 (*c*)). The disconnected numerals are processed individually in the pre-processing and recognition stages.

3. Pre-processing

In smoothing a numeral pattern only the corner points like C in Fig. 3 (*c*) are removed. Now the handwritten numerals consisting of curves usually do not have a vertical line segment particularly in the lower half and are classified in group 1. The rest consisting mainly of horizontal and vertical strokes form group 2.

3.1. *Border extraction*

First the border extraction for group 1 numerals is discussed. The point on the border from which the extraction is to start is found first. For this the algorithm starts from the top right point (L in Fig. 4 (*a*)) of the numeral pattern and traces the border points in clockwise direction. The corresponding directional codes are noted.

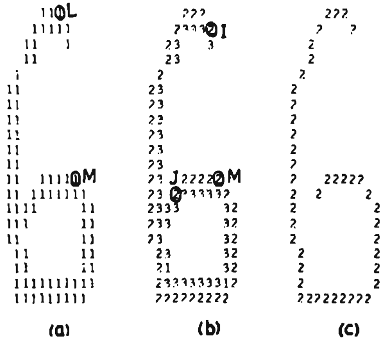

Figure 4. The numeral pattern '6' at different stages during pre-processing.

The algorithm runs as follows :

Step 1. If the directional code is 3, note the position co-ordinates of the corresponding point M, find the next directional code and go to step 2. Otherwise, find the next directional code and repeat step 1.

Step 2. If the directional code is 3 or 4, find the next directional code and repeat step 2. If it is 5, go to step 3. Otherwise, find the next directional code and go to step 1.

Step 3. M is the starting point for border extraction (Fig. 4 (*a*)).

In fact, M is the starting point of the first subpattern q_1 (Fig. 5) if the numeral pattern is scanned from above right in clockwise direction. It can be seen that the subpattern q_1 must be present in every group 1 numeral pattern. Now the extraction should be done only from one side of the border (thick lines in Fig. 1). For this when one side of the border is extracted, extraction of the other side is blocked. This is done by changing the gray levels of the pixels on the other side of the border to 3 from 1. The gray levels of the other border pixels are changed to 2.

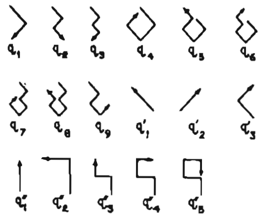

Figure 5. Seventeen subpatterns for numerals.

The algorithm for the formation of the string of directional codes from border points is as follows :

Step 1. Let S = M.

Step 2. S's next point, say T, on the border is found. The directional code is, say, d. The level of S is made 2. The directional codes $d+1$, $d+2$, $d+3$ (all mod 8) are calculated. In these directions from S, the end points on the other side of the border are found and whether the end points are ' very far ' from S is checked. (By ' very far ' is meant a distance of more than 3 units where a unit is the size of a pixel. Here 3 units is the maximum thickness of the pattern.) The levels of the end points that are not very far are changed to 3 in case they are 1, and left unchanged otherwise. The end points that are very far are ignored. If the level of T is more than 1, go to step 4. Otherwise, go to step 3. For example, in Fig. 6 the point S has the directional code $d=4$. Corresponding end points in directions 5, 6, 7 are L, M, N of which only L is ' very far ' from S.

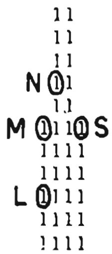

Figure 6.

Step 3. The directional code *d* is taken into the output string sequentially. Make S = T. Go to step 2.

Step 4. T's next three points on the border are found. If the level of any of these three points is 1, go to step 2. Otherwise, go to step 5.

Step 5. Formation of one string is over. The string is stored. In order to find the starting point of the second string the algorithm again moves along the border from T until a point S on the border is found to have level 1. Formation of the second string (which may happen to be null) will start from S. Go to step 2. For example, in Fig. 4 (*b*) the first string ends at point I. The string is

3344444566666666668887888888888811811223

The next string starts at J.

The algorithm terminates when the starting point (M in Fig. 4) is encountered again. The second string in the example is 112222. The ultimate skeleton extracted by the algorithm is shown in Fig. 4 (*c*) whose one-dimensional form is given by the two strings of directional codes mentioned above.

The border extraction algorithm for group 2 numerals is the same as the one described for group 1 numerals except for the following considerations :

(i) Unlike the earlier case the starting point for string formation here is fixed. It is the extreme lower left point of the vertical line segment (M in Fig. 7). From this point the algorithm moves upwards in clockwise direction.

(ii) The algorithm terminates before reaching the starting point. Its termination point (N in Fig. 7) is near the top of the vertical line segment and is identified beforehand, along with the vertical line segment. In the earlier case both starting and termination points were the same. For example, the output string for the extracted border in Fig. 7 is

88188188176666666678818181112

The noise of the type shown in Fig. 8 is removed during border extraction.

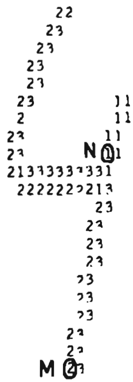

Figure 7. The numeral pattern ‘4’ after border extraction.

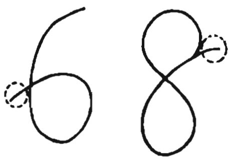

Figure 8. Noise is shown within dotted circles.

4. Recognition

The output of the pre-processing algorithm in the form of one string or two strings of directional codes is the input of the recognition procedure. Here first a subpattern is recognized on the basis of each such string and then the whole numeral pattern is recognized.

4.1. *Recognition of subpatterns*

Recognition of subpatterns in the case of group 1 numerals is first discussed. Of the two strings of directional codes the first one is recognized either as a null subpattern (q_0 or q_a) or as one of the 9 subpatterns $q_1, \ldots, q_9$ (Fig. 5) by an automaton which is defined as

$$A = (D, Q, \sigma, q_0, q_a)$$

where

D is the set of directional codes $= \{1, 2, \ldots, 8\}$
Q is the set of subpatterns $= \{q_0, q_a, q_1, \ldots, q_9\}$
σ is a mapping of $Q \times D$ into Q
q_0 is the initial subpattern
q_a is the null subpattern

The interpretation of $\sigma(q, d) = q'$; $q, q' \in Q$ and $d \in D$ is that the subpattern q after the addition of the directional code d becomes the subpattern q'. σ is given in Table 1 where

$$\sigma(q_a, d) = q_a \quad \text{for all } d \in D$$

$$\sigma(q_0, d) = \begin{cases} q_0 \quad \text{or} \quad q_a & \text{for all } d \neq 5 \\ q_2 & \text{for } d = 5 \end{cases}$$

σ	1	2	3	4	5	6	7	8
q_0	q_a	q_0	q_0	q_0	q_1	q_a	q_a	q_a
q_1	q_4	q_a	q_2	q_1	q_1	q_1	q_1	q_1
q_2	q_9	q_3	q_2	q_3	q_3	q_a	q_a	q_a
q_3	q_7	q_a	q_a	q_3	q_3	q_3	q_3	q_3
q_4	q_4	q_4	q_a	q_a	q_a	q_a	q_5	q_4
q_5	q_6	q_a	q_a	q_a	q_a	q_5	q_5	q_5
q_6	q_6	q_6	q_6	q_6	q_6	q_6	q_a	q_a
q_7	q_7	q_7	q_7	q_7	q_7	q_a	q_8	q_7
q_8	q_8	q_8	q_8	q_8	q_8	q_8	q_8	q_8
q_9	q_9	q_9	q_a	q_a	q_a	q_a	q_a	q_9
q_a	q_a	q_a	q_a	q_a	q_a	q_a	q_a	q_a

Table 1.

It is clear that once the automaton reaches the subpattern q_a, it will end up with q_a. Now, for any string $d_1 \ldots d_n$ of directional codes and for any $q \in Q$, $\sigma(q, d_1 \ldots d_n)$ is defined as $\sigma(\sigma(q, d_1), d_2 \ldots d_n)$. If $d_1 \ldots d_n$ is one input (first) string for the recognition algorithm, it is recognized as the subpattern $q \in Q$ where $\sigma(q_0, d_1 \ldots d_n) = q$. If the second string is null, then it is recognized as a null subpattern q'_0. Otherwise, it is recognized as one of the three subpatterns q'_1, q'_2, q'_3 (Fig. 5) by another automaton

$$A' = (D, Q', \sigma', q'_0)$$

where

$Q' = \{q'_0, q'_1, q'_2, q'_3\}$
σ' is a mapping of $Q' \times D$ into Q'
q'_0 is the initial subpattern

σ' is given in Table 2. For any string of directional codes $d_1, \ldots, d_n$ and for any $q' \in Q'$, $\sigma'(q', d_1 \ldots d_n)$ is similarly defined. The second string $d_1 \ldots d_n$ is recognized as the subpattern $q' \in Q'$ if $\sigma'(q'_0, d_1 \ldots d_n) = q'$.

Recognition of subpatterns for group 2 numerals is similarly done. Here possible subpatterns are $q''_1, \ldots, q''_5$ (Fig. 5) and the automaton is

$$A'' = (D, Q'', \sigma'', q''_0, q''_a)$$

σ'	1	2	3	4	5	6	7	8
q'_0	q'_2	q'_0	q'_0	q'_0	q'_0	q'_0	q'_1	q'_0
q'_1	q'_3	q'_a	q'_a	q'_a	q'_a	q'_1	q'_1	q'_1
q'_2	q'_2	q'_2	q'_2	q'_2	q'_2	q'_2	q'_2	q'_2
q'_3	q'_3	q'_3	q'_3	q'_3	q'_3	q'_3	q'_3	q'_3
q'_a	q'_a	q'_a	q'_a	q'_a	q'_a	q'_a	q'_a	q'_a

Table 2.

σ''	1	2	3	4	5	6	7	8
q''_0	q''_0	q''_a	q''_a	q''_a	q''_a	q''_a	q''_0	q''_1
q''_1	q''_1	q''_a	q''_a	q''_a	q''_2	q''_2	q''_1	q''_1
q''_2	q''_3	q''_a	q''_a	q''_a	q''_2	q''_2	q''_2	q''_3
q''_3	q''_3	q''_4	q''_4	q''_a	q''_a	q''_a	q''_3	q''_3
q''_4	q''_4	q''_4	q''_4	q''_5	q''_5	q''_a	q''_a	q''_a
q''_5	q''_a	q''_a	q''_5	q''_5	q''_5	q''_5	q''_5	q''_5
q''_a	q''_a	q''_a	q''_a	q''_a	q''_a	q''_a	q''_a	q''_a

Table 3.

where

$Q'' = \{q''_0, q''_a, q''_1, \ldots, q''_5\}$
σ'' is a mapping of $Q'' \times D$ into Q''
q''_0 is the initial subpattern
q''_a is the null subpattern

σ'' is given in Table 3. For any string of directional codes $d_1, \ldots, d_n$ and for any $q'' \in Q''$, $\sigma''(q'', d_1 \ldots d_n)$ is similarly defined. An input string $d_1 \ldots d_n$ is recognized as the subpattern $q'' \in Q''$ if $\sigma''(q''_0, d_1 \ldots d_n) = q''$.

4.2. *Recognition of numeral patterns*

For a group 1 numeral one or two subpatterns have been recognized while for a group 2 numeral only one subpattern has been recognized. The procedure for recognition of a numeral pattern on the basis of these subpatterns is given in Tables 4 and 5. The combinations of subpatterns which are not in these tables are not recognized as numerals. The four cases of multiplicity of numeral patterns in these tables are settled through thresholding on the basis of certain observations. In the sixth row of Table 4 the subpatterns q_3 and q'_3 form either ' 9 ' or ' 3 ' (Fig. 9). In order to discriminate between the two patterns the ratio

$$t_1 = \frac{\text{length of the first part of } q_3}{\text{length of the second part of } q_3 + \text{length of } q'_3}$$

Subpatterns 1st	2nd	Numeral patterns
q_1	q'_2	5
q_1	q'_3	9
q_2	q'_0	2
q_2	q'_3	2
q_3	q'_0	3
q_3	q'_3	9 or 3
q_4	q'_0	0 or 6
q_4	q'_2	5
q_5	q'_0	8
q_6	q'_2	8
q_7	q'_0	3
q_8	q'_0	8
q_9	q'_0	2
q_9	q'_3	2

Table 4. For group 1 numerals.

Subpatterns	Numeral patterns
q''_1	1
q''_2	1 or 7
q''_3	4
q''_4	4 or 9
q''_5	9

Table 5. For group 2 numerals.

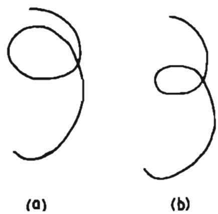

Figure 9. (*a*) Handwritten '9'. (*b*) Handwritten '3'.

is calculated. If the value of t_1 is sufficiently small, the numeral pattern is recognized as '9'. Otherwise it is recognized as '3'. The optimum threshold value of t_1 has been found to be 0·33 from our data base. Another case of ambiguity is the seventh row of Table 4 where the numeral pattern is either '0' or '6'. The ratio

$$t_2 = \frac{\text{length of the first part of } q_4}{\text{length of the second part of } q_4}$$

is calculated. If the value of t_2 is small enough, the numeral pattern is '6', otherwise it is '0'. The optimum threshold value of t_2 is 0·75. In the case of the second row of Table 5, if the ratio

$$t_3 = \frac{\text{length of the horizontal part of } q''_2}{\text{length of its vertical part}}$$

is less than 0·30, the pattern is recognized as '1'. Otherwise, it is recognized as '7'. In the fourth row of Table 5 the numeral pattern is either '4' or '9'. The discrimination is made on the basis of the value of the ratio

$$t_4 = \frac{\text{total length of the middle horizontal and vertical parts of } q''_4 - \text{length of the top horizontal part}}{\text{length of the bottom vertical part}}$$

If it is less than 1·33, the numeral pattern is '9', otherwise it is '4'.

5. Results and discussion

In the present method input handwritten numerals can in general be unconstrained in shape and size. But in the same chain of connected numerals, the size and base line of individual numerals should not vary much. The thickness of handwritten numerals may vary from one part to another. But the present algorithm demands that the thickness be nearly uniform. The maximum thickness is a parameter of the border extraction algorithm. In our present data base the maximum thickness is three units. However, it may

Figure 10. A sample of handwritten numerals which were correctly classified.

be possible to develop an algorithm to find the maximum thickness of a pattern if the value is not supplied. Our recognition method is invariant under the size of the input pattern provided all the numerals present in one input chain of connected numerals are of the same height. A sample of individual numeral patterns that were correctly recognized is shown in Fig. 10.

The recognition rate on the testing data is about 90%. Programs have been written in Fortran IV for segmentation, smoothing, border extraction and recognition. The processing speed is about one numeral per second in EC-1033. The speed can be improved if the programs are written in an assembly language.

ACKNOWLEDGMENTS

The authors gratefully acknowledge the help of Mr. J. Gupta and Mr. S. Chakraverti in preparing the manuscript.

REFERENCES

ALI, F., and PAVLIDIS, T., 1977, *I.E.E.E. Trans. Syst. Man Cyber.*, **7**, 537.

FOCHT, L. R., and BURGER, A., 1976, *Proc. I.E.E.E. Int. Conf. Cyber. and Society*, p. 489.

FU, K. S., 1974, *Syntactic Methods in Pattern Recognition* (New York: Academic Press); 1977, *Applications of Syntactic Pattern Recognition* (New York: Springer).

HIMMEL, D. P., 1976, *Proc. I.E.E.E. Int. Conf. Cyber. and Society*, p. 479.